

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-021104

(43)Date of publication of application : 23.01.1998

(51)Int.Cl.

G06F 11/18

(21)Application number : 08-168804

(71)Applicant : FUJITSU LTD

(22)Date of filing : 28.06.1996

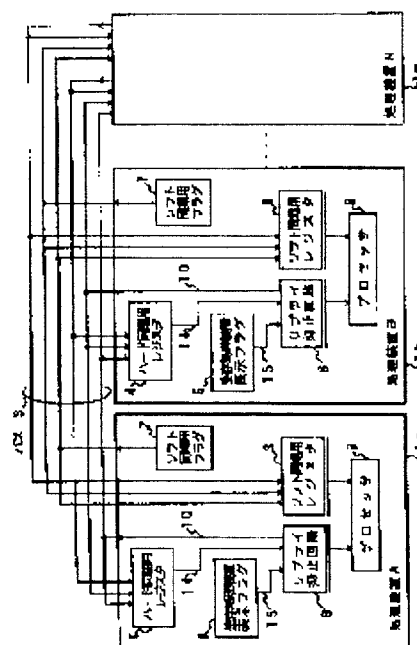
(72)Inventor : ISHIZUKA KOJI
 TAKENO TAKUMI
 KATO SHINYA
 TAKESHITA KATSUNORI
 SUGANO FUMITAKE
 GOTO SEIJI

(54) INFORMATION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To execute the hard synchronous processing of a clock level by means of a software by providing a simple hardware circuit concerning an information processor which comprises plural processors executing a same processing operation simultaneously.

SOLUTION: The information processor is provided with a replay suppressing circuit 6 for suppressing a replay as against the access of a hard synchronous register 4 till the whole processors perform access to the hard synchronous register 4 and the replay is simultaneously returned when the whole processors perform access to the hard synchronous register 4. The rough soft synchronization of an instruction level is obtained through the use of a soft synchronous flag 7 and a soft synchronous register 8.



LEGAL STATUS

[Date of request for examination]

30.07.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3241997

[Date of registration]

19.10.2001

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-21104

(43) 公開日 平成10年(1998) 1月23日

(51) Int.Cl.⁶

G 0 6 F 11/18

識別記号

3 1 0

庁内整理番号

F I

G 0 6 F 11/18

技術表示箇所

3 1 0 F

審査請求 未請求 請求項の数8 ○L (全 8 頁)

(21) 出願番号 特願平8-168804

(22) 出願日 平成8年(1996) 6月28日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 石塚 孝治

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 竹野 巧

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 小笠原 吉義 (外2名)

最終頁に続く

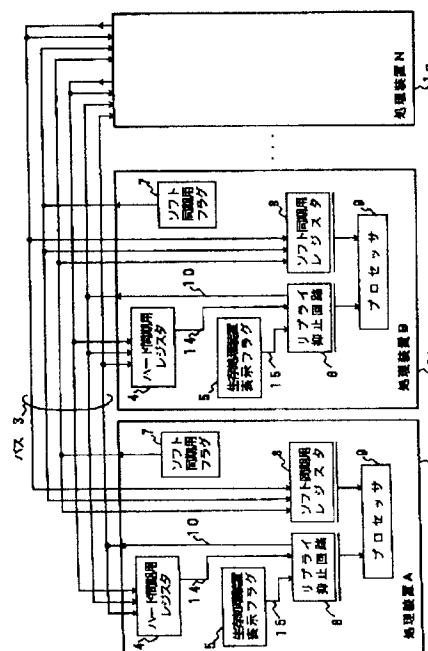
(54) 【発明の名称】 情報処理装置

(57) 【要約】

【課題】 同時に同じ処理動作を行う複数の処理装置からなる情報処理装置に関し、簡単なハードウェア回路を設けることにより、ソフトウェアによるクロックレベルのハード同期処理を可能にすることを目的とする。

【解決手段】 ハード同期用レジスタ4のアクセスに対するリブライを、全処理装置がハード同期用レジスタ4にアクセスするまで抑止するリブライ抑止回路6を設け、全処理装置がハード同期用レジスタ4にアクセスしたときに同時にリブライを返す。また、ハード同期の前にソフト同期用フラグ7とソフト同期用レジスタ8を用いて、命令レベルの粗いソフト同期をとる。

本発明の原理説明図



【特許請求の範囲】

【請求項1】 バスを介して接続され同時に同じ処理動作を行う少なくとも3台の処理装置を有し、これらの処理装置の出力結果を比較する回路を持つ情報処理装置において、前記各処理装置は、ハード同期用レジスタと、前記ハード同期用レジスタに対するアクセスのリブライ保留中を示す信号線と、前記ハード同期用レジスタのアクセスに対するリブライを、全処理装置の前記信号線がリブライ保留中を示すまで抑止するリブライ抑止回路とを備え、ハード同期用レジスタのアクセスに対するリブライによって、その後の各処理装置の動作をクロックレベルで同期させるようにしたことを特徴とする情報処理装置。

【請求項2】 請求項1記載の情報処理装置において、前記ハード同期用レジスタは、リードアクセスのみ可能であることを特徴とする情報処理装置。

【請求項3】 請求項1または請求項2記載の情報処理装置において、前記リブライ抑止回路の内部または外部の付加回路として、前記ハード同期用レジスタのアクセス受信から所定の時間経過した場合に、強制的にリブライを返すためのタイムアウト検出回路を有することを特徴とする情報処理装置。

【請求項4】 請求項3記載の情報処理装置において、前記タイムアウト検出回路は、前記ハード同期用レジスタのリードアクセスによってタイムアウト検出のためのカウントを開始するカウンタを有し、かつ、該カウンタのカウント開始後に前記ハード同期用レジスタを用いた同期合わせに参加する処理装置が増えるごとに、該カウンタのカウント値をクリアして再度カウントし直す制御を行う回路を有することを特徴とする情報処理装置。

【請求項5】 請求項1記載の情報処理装置において、前記各処理装置が正常に動作している処理装置であるかどうかを示す生存処理装置表示手段を有し、該生存処理装置表示手段が正常に動作していない処理装置の存在を示す場合に、その正常に動作していない処理装置のリブライ保留中を示す信号線を、リブライ抑止解除の条件から外す回路を有することを特徴とする情報処理装置。

【請求項6】 請求項3、請求項4または請求項5記載の情報処理装置において、前記ハード同期用レジスタのリードアクセスに対して、ハード同期に成功または失敗した処理装置を判別する情報をリードデータとして返すようにしたことを特徴とする情報処理装置。

【請求項7】 請求項1記載の情報処理装置において、前記各処理装置は、ソフト同期用フラグと、全処理装置の前記ソフト同期用フラグと処理装置分の信号線を用いて接続されているソフト同期用レジスタと、前記ハード同期用レジスタを用いたハード同期処理の前に、自処理装置の前記ソフト同期用フラグをセットし、その後前記ソフト同期用レジスタをリードしながら全処理装置が前記ソフト同期用フラグをセットするのを待ち、全処理

装置が前記ソフト同期用フラグをセットしたのを確認してから前記ハード同期処理に移行するソフト同期処理手段とを有することを特徴とする情報処理装置。

【請求項8】 請求項7記載の情報処理装置において、前記ソフト同期処理手段は、前記ソフト同期用フラグのライトデータとして、“1”および“0”の両方を使用し、2回以上のソフト同期動作を行うことを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、同時に同じ処理を行う少なくとも3台の処理装置を有する高信頼性情報処理装置に関する。

【0002】近年、情報処理装置が各種分野で広く利用されるようになってきたのに伴い、情報処理装置が故障した場合に、社会や経済に与える影響も大きくなってきている。このため、できる限り故障が発生しにくく、また万一故障が発生した場合にも確実に故障が検出でき、さらには処理装置が停止せずに、処理内容の整合性を保ちながら処理を実行できるような信頼性の高い情報処理装置が求められている。

【0003】

【従来の技術】図7は、従来の高信頼性情報処理装置の例を示す。処理装置100a～処理装置100cは、バス130で接続され、同時に同じ処理を実行する。バス130に流れる処理結果を比較し、処理結果が一致しなければ、処理結果の多数決論理により、他の処理装置の処理結果と一致しない処理結果を出した処理装置を故障と判断して、その処理装置をバス130から切り離す。

【0004】この故障の検出のために、具体的には例えば処理装置100aがマスタとなって、以下のような手順により、処理結果の比較を行う。まず、各処理装置100a～100cは、バス130に流れる処理要求のデータを入力し（図7の ）、各々処理回路110によって要求されたデータ処理を同時に実行する（図7の ）。バス130へのデータ出力は、マスタとなっている処理装置100aだけが行う。マスタとなっている処理装置100aがデータの処理結果をバス130に流すと（図7の ）、他の処理装置100b、100cは、それぞれその処理結果を自装置内の故障検出回路120に入力する。

【0005】一方、処理装置100b、100cの処理回路110は、先のデータ入力に対する処理結果をバス130に流すことなく、自装置の故障検出回路120に出力する（図7の ）。

【0006】各処理装置100b、100cにおける故障検出回路120は、バス130から入力した処理装置100aの処理結果と自装置で処理した処理結果とを比較し、その比較結果を相互に送受する（図7の ）。

【0007】処理結果を比較した結果、処理結果がすべ

10

20

30

40

50

3

て一致すれば、すべての処理装置100a~100cが正常であると判断することができる。もし、処理装置100b、100cのいずれか一方の故障検出回路120による比較結果が不一致であれば、その処理装置100bまたは処理装置100cが故障ということになる。処理装置100b、100cの故障検出回路120による比較結果が双方とも不一致であれば、処理装置100aが故障したと判断する。

【0008】故障と判断された処理装置は、バス130から自動的に切り離され、残りの2台の処理装置によって同時動作による処理が継続される。このような高信頼性情報処理装置は、各処理装置100a~100cが同時に同じ処理を実行し、その処理結果を比較するため、全処理装置がクロックレベルで同期して動作する必要がある。従来、各処理装置100a~100cをクロックレベルで同期させる方法としては、同じクロック信号140を各処理装置100a~100cへ与えるとともに、リセット信号150を共通に与えて、リセットを同時に解除することにより、クロックに基づく動作開始を合わせる方法が用いられている。

【0009】

【発明が解決しようとする課題】以上のような従来の高信頼性情報処理装置は、次のような問題点があった。各処理装置の動作をクロックレベルで同期させるときに、リセットの同時解除で行っているため、何らかの理由によりリセット解除が各処理装置でずれた場合に、ソフトウェアによってハード同期をとることができなかった。

【0010】本発明は上記問題点の解決を図り、簡単なハードウェア回路を設けることにより、ソフトウェアによるハード同期処理の可能な高信頼性情報処理装置を提供することを目的とする。

【0011】

【課題を解決するための手段】図1は、本発明の原理説明図である。図1において、1a~1nは同一クロックにより同時に同じ処理を実行する多重化構成の処理装置、3はデータ線と各種制御信号線からなるバス、4はリード専用のハード同期用レジスタ、5はリード/ライト可能な生存処理装置表示フラグ、6はハード同期用レジスタ4に対するアクセスのリブライを抑制するリブライ抑止回路、7はリード/ライト可能なソフト同期用フラグ、8はリード専用のソフト同期用レジスタ、9はソフトウェアに基づく命令を実行するプロセッサ、10はリブライ保留中を示すリブライ抑止回路ビジー信号、14はハード同期用レジスタ出力信号、15は生存処理装置表示信号を表す。

【0012】ハード同期用レジスタ4~プロセッサ9は、すべての処理装置1a~1nの中に存在する。ハード同期用レジスタ4には、各処理装置1a~1nに設けられたリブライ抑止回路6のビジー信号10が接続されており、リブライ抑止回路6には、自処理装置内のハー

4

ド同期用レジスタ4と生存処理装置表示フラグ5の値とが入力される。また、ソフト同期用レジスタ8には、各処理装置1a~1nのソフト同期用フラグ7の値が入力信号として接続されている。

【0013】本発明では、処理装置1a~1nのソフトウェアがソフト同期用フラグ7をセット後、ソフト同期用レジスタ8をリードしながら全処理装置1a~1nがソフト同期用フラグ7をセットするのを待つことで、ソフトウェアによるクロックのおおまかな同期を可能とする。

【0014】次に、処理装置1a~1nのソフトウェアがハード同期用レジスタ4を1回リードすることで、ハード同期を可能とする。また、以上の同期合わせの処理により、同期の合わない故障処理装置の検出も可能とする。

【0015】具体的には、以下のとおりである。ハード同期のために、ハード同期用レジスタ4を設けるとともに、ハード同期用レジスタ4に対するプロセッサ9からのアクセスのリブライ保留中を示す信号線を設け、ハード同期用レジスタ4のアクセスに対するリブライを、全処理装置1a~1nの信号線がリブライ保留中(リブライ抑止回路ビジー信号10)を示すまでリブライ抑止回路6によってリブライを抑制する。

【0016】例えば、何らかの理由によりリセット解除が各処理装置1a~1nでずれたような場合にも、ソフトウェアによってハード同期用レジスタ4をアクセスすれば、そのリブライが各処理装置1a~1nに同時に返るので、その後の各処理装置1a~1nの動作をクロックレベルで同期させることができる。

【0017】特に、ハード同期用レジスタ4をリード専用としてリードアクセスのみ可能とすることにより、ハード量の必要以上の増加を防ぐことができる。また、リブライ抑止回路6の内部または外部に、ハード同期用レジスタ4のアクセス受信から所定の時間経過した場合に、強制的にリブライを返すタイムアウト検出回路(図示省略)を設ける。これにより、例えば故障処理装置が原因でリブライ抑止が永久に解除されなくなるというような事態を回避することが可能になる。

【0018】さらに、タイムアウト検出回路を、ハード同期用レジスタ4のリードアクセスによってタイムアウト検出のためのカウントを開始するカウンタによって構成し、このカウンタのカウント開始後にハード同期用レジスタ4を用いた同期合わせに参加する処理装置が増えるごとに、このカウンタのカウント値をクリアして再度カウントし直す制御を行う回路を設ける。

【0019】こうすることによって、ハード同期終了後に、故障していない生存処理装置の同期がずれたままになってしまうことを防ぐことができる。また、処理装置1a~1nの各々が正常に動作している処理装置であるかどうかを示す生存処理装置表示フラグ5を設ける。そ

5

して、リブライ抑止回路6では、生存処理装置表示フラグ5が正常に動作していない処理装置の存在を示す場合に、その正常に動作していない処理装置のリブライ保留中を示す信号線を、リブライ抑止解除の条件から外す。

【0020】これによって、故障があらかじめ判明している処理装置をハード同期の対象とすることにより同期合わせの時間が長くなるのを防ぐことが可能になる。また、ハード同期用レジスタ4のリードアクセスに対して、ハード同期に成功または失敗した処理装置を判別する情報を、ハード同期用レジスタ4のリードデータとして返すようにする。これによって、ハード同期に失敗した故障処理装置をソフトウェアが簡単に認識できるようになり、故障処理装置の切り離しを自動的に行うことが可能になる。

【0021】また、ハード同期の前に行うソフト同期のために、各処理装置1a～1nごとに、ソフト同期用フラグ7と、全処理装置1a～1nのソフト同期用フラグ7と処理装置分の信号線を用いて接続されているソフト同期用レジスタ8と、自処理装置のソフト同期用フラグ7をセットし、その後にソフト同期用レジスタ8をリードしながら全処理装置がソフト同期用フラグ7をセットするのを待ち、全処理装置1a～1nがソフト同期用フラグ7をセットしたのを確認してから、ハード同期用レジスタ4を用いたハード同期処理に移行するソフト同期処理手段とを設ける。

【0022】これにより、例えばハード同期におけるタイムアウト値より大きなずれを持つ処理装置の同期を、事前にソフト同期によってハード同期の可能な範囲内になるように合わせることが可能になる。

【0023】また、ソフト同期処理手段は、ソフト同期用フラグ7のライトデータとして、“1”および“0”の両方を使用し、2回以上のソフト同期動作を行うようにする。これによって、ソフト同期用フラグ7の出力として“1”または“0”の一方の固定故障が生じた場合に、その固定故障を検出することが可能になる。

【0024】

【発明の実施の形態】次に、図2ないし図6を用いて、本発明の実施の形態における回路およびソフトの動作を、順次詳細に説明する。

【0025】図2は、本発明の実施の形態によるハード同期のための回路構成図である。図中、11はハード同期完了検出回路、12はハード同期用レジスタ値変化検出回路、13はカウンタによって構成されるタイムアウト検出用タイマ、16はハード同期完了信号、17はタイムアウト信号、18はタイマリセット信号、19はリブライ抑止解除信号、20はプロセッサ9によるハード同期用レジスタ4のリードコマンド、21はプロセッサ9へのハード同期用レジスタ4のリードリブライを表す。なお、図1と同符号のものは図1に示すものに対応するが、図3に示すリブライ抑止回路6にハード同期完

6

了検出回路11、ハード同期用レジスタ値変化検出回路12、タイムアウト検出用タイマ13を含めたものを、図1ではリブライ抑止回路6として簡略化して表している。

【0026】以下では、処理装置が3台（処理装置A、B、C）であるとして説明する。ハード同期用レジスタ4と生存処理装置表示フラグ5は、それぞれ各処理装置に対応するフラグ（#1～#3）を持つ。生存処理装置表示フラグ5の値は、各処理装置1a～1cが実装されているか否かによって自動的に設定される。また、故障を検出したときに、生存処理装置表示フラグ5の値をソフトウェアが設定してもよい。

【0027】ハード同期完了検出回路11は、各処理装置ごとのハード同期用レジスタ出力信号14と生存処理装置表示信号15を反転した信号とのオア（論理和）回路と、その全オア回路の出力信号のアンド（論理積）回路とからなる。

【0028】ハード同期用レジスタ値変化検出回路12は、各処理装置ごとのハード同期用レジスタ出力信号14と、フリップフロップFFによって保持した1クロック前のハード同期用レジスタ出力信号14との排他的論理和回路と、それらの排他的論理和回路の出力信号のオア（論理和）回路とからなる。

【0029】プロセッサ9がハード同期用レジスタ4に対しリードコマンド20によるリード動作を行うと、リブライ抑止回路6のビジー信号10がオンとなる。リブライ抑止回路6のビジー信号10は、処理装置1a～1cのハード同期用レジスタ4の入力信号となっている。このとき、少なくともハード同期用レジスタ4における自処理装置の該当ビットは、“0”から“1”へ値が変化しているため、ハード同期用レジスタ値変化検出回路12によって、ハード同期用レジスタ4のアクセスのタイムアウト検出用タイマ13に対し、タイマリセット信号18が発行される。

【0030】また、ハード同期に参加する（ハード同期用レジスタ4をリードする）処理装置が増えるときにも、ハード同期用レジスタ値変化検出回路12によってタイムアウト検出用タイマ13に対し、タイマリセット信号18が発行される。

【0031】タイマリセット信号18を受け取ったタイムアウト検出用タイマ13は、“0”にリセットされる。タイムアウト検出用タイマ13は、タイマ値が最大となったところで、タイムアウト信号17をオンとする。なお、この例ではタイムアウト検出用タイマ13をアップカウンタによって構成した場合について説明しているが、ダウンカウンタによって構成してもよいことは言うまでもない。

【0032】ハード同期用レジスタ4の出力信号14は、生存処理装置表示フラグ5の出力信号である生存処理装置表示信号15とともにハード同期完了検出回路1

1に入力され、そこで全生存処理装置のリブライ抑止回路ビジー信号10がオンとなったことを条件として、ハード同期完了信号16が生成される。

【0033】ハード同期完了信号16とタイムアウト信号17のオア信号がリブライ抑止解除信号19となり、この信号を受信した時点で、リブライ抑止回路6は、そのときのハード同期用レジスタ4の値をリードデータとして、プロセッサ9に対してリードリブライ21を発行する。

【0034】以上のような回路動作により、プロセッサ9は、ハード同期用レジスタ4をリードするだけで、タイムアウト検出用タイマ13のタイマ値の最大値より小さなずれの処理装置をハード同期させることが可能になる。また、ハード同期用レジスタ4のリードデータにより、ハード同期に成功または失敗した処理装置がどれであるかを判別することができる。ハード同期用レジスタ4のリードデータのうち、ビット値が“0”になっている処理装置が、ハード同期に失敗した処理装置である。

【0035】図3は、図2に示すリブライ抑止回路の説明図である。リブライ抑止回路6は、図3に示すようにリードデータ6-1の保持回路と抑止回路ビジー信号生成回路6-2とを持つ。プロセッサ9からリードコマンド20を受け取ると、抑止回路ビジー信号生成回路6-2は、リブライ抑止解除信号19を受信するまで、抑止回路ビジー信号10をオンにし、リードリブライ21の発行を抑止する。

【0036】リブライ抑止解除信号19を受信すると、抑止回路ビジー信号10をオフに戻し、ハード同期用レジスタ4からの出力信号14をリードデータ6-1として、リードリブライ21をプロセッサ9へ返す。

【0037】図4は、本発明の実施の形態によるソフト同期のための回路構成図である。図中の22はソフト同期用フラグ7の出力信号、23はプロセッサ9によるソフト同期用レジスタ8のリード動作（リードコマンド）、24はプロセッサ9へのソフト同期用レジスタ8のリードリブライを表す。

【0038】プロセッサ9が、ソフト同期用フラグ7に“1”または“0”をライトすると、それに対応するソフト同期用レジスタ8のビットにその値が反映される。すなわち、例えば処理装置1aがソフト同期用フラグ7に“1”をライトした場合には、ソフト同期用レジスタ8のビット#1が“1”となる。

【0039】プロセッサ9は、ソフト同期用レジスタ8を監視し、全ての処理装置1a～1cがソフト同期用フラグ7にライトするまで待つことにより、ソフト同期が可能となる。また、この動作をソフト同期用フラグ7にライトするデータ値を変えて行うことにより、故障処理装置の検出が可能となる。

【0040】図5は、本発明を用いたソフトウェアの処理フローチャートである。最初に生存処理装置表示フラ

グ5を、処理装置の実装状況に合わせてセットする（ステップS1）。次に、ソフト同期用フラグ7に“1”をライトし（ステップS2）、ソフト同期用レジスタ8をリードしながら、生存処理装置表示フラグ5にセットした処理装置分のビットがすべて“1”となるまで、ループして待つ（ステップS3～S5）。

【0041】このとき、故障処理装置発生によって無限ループに陥らないようにするために、ループ回数のタイムアウト値n1を設定しておく。従って、全処理装置がソフト同期用フラグ7をセットするか、ループ回数がタイムアウト値n1になったときにこのループを抜け出し、次のステップS6へ移る。

【0042】ステップS6では、各処理装置のソフト同期用フラグ7のリードタイミングを調整するため、無効（NOP）命令の発行などにより少なくとも1命令だけWaitした後、次のステップS7へ進む。

【0043】ステップS7では、上記のソフト同期の結果に基づき、生存処理装置表示フラグ5を再セットする。すなわち、ソフト同期用フラグ7をセットしない状態でタイムアウトになった処理装置があれば、生存処理装置表示フラグ5にその処理装置が故障した旨をセットする。

【0044】次に、例えばハードウェアの故障によってソフト同期用フラグ7が“1”に固定されていた処理装置があった場合、ステップS2～S5の処理だけでは、故障を判別できないので、ソフト同期用フラグ7にライトするデータを“0”にして、同様の処理を行う（ステップS8～S12）。

【0045】以上の2回のソフト同期処理で、生存処理装置は命令レベルの粗い同期がとれたことになるので、次にハード同期用レジスタ4のリード動作によって、生存処理装置をクロックレベルで完全にハード同期させる（ステップ13）。また、ハード同期用レジスタ4のリードデータの結果から、生存処理装置表示フラグ5の再セットをおくことにより（ステップS14）、その後の処理に生存処理装置表示フラグ5を利用することができるようにする。

【0046】図6は、以上のソフト同期とハード同期のタイムチャートを示している。処理装置A、B、Cが同期処理を行うものとする。最初に処理装置Aが動作し、ソフト同期用フラグ（SSF）7に“1”をセットした後、ソフト同期用レジスタ（SSR）8をリードして、すべてが“1”になるのを待っている。処理装置Bも同様に、ソフト同期用フラグ7に“1”をセットして、ソフト同期用レジスタ8をリードする。一番遅く、処理装置Cがソフト同期用フラグ7に“1”をセットすると、ソフト同期用レジスタ8のビットがすべて“1”になるので、ソフト同期の確認がとれたことになる。固定障害の検出のため、ソフト同期用フラグ7に“0”をセットして同様にソフト同期を繰り返す。図6に示すように、

ソフト同期後のずれは、ソフト同期前のずれよりも小さくなる。

【0047】その後、各処理装置A～Cがハード同期用レジスタ(HSR)4をリードすれば、それぞれの処理装置A～Cにリードブライが返った時点で、ハード同期が完全にとれていることになる。

【0048】ハード同期のみで大きなずれの同期合わせをサポートしようとすると、ハード量が増加することになるが、ソフト同期により粗い同期をとった後に、ハード同期の処理を行うことにより、大きな同期ずれが生じている場合でも、簡単に同期合わせすることができるようになる。

【 0 0 4 9 】

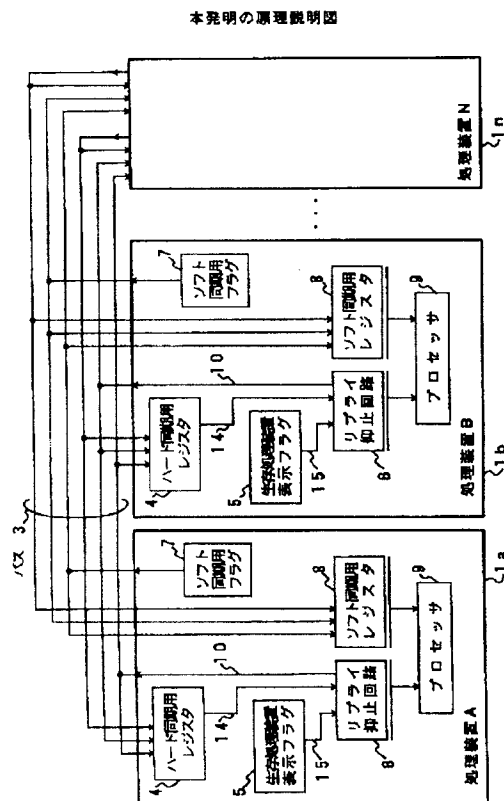
【発明の効果】以上説明したように、本発明によれば、例えばリセットがずれて同期がずれてしまった処理装置を、簡単な回路とソフトウェアの処理により、ハード同期をとることが可能になる。また、同期合わせの際に、故障処理装置の検出も可能になり、高信頼性情報処理装置の信頼性向上に寄与するところが大きい。

【図面の簡単な説明】

【図 1】 本発明の原理説明図である。

【図 2】 本発明の実施の形態によるハード同期のための回路構成図である。

【図 1】



【図3】図2に示すリプライ抑止回路の回路説明図である。

【図４】本発明の実施の形態によるソフト同期のための回路構成図である。

【図5】本発明を用いたソフトウェアの処理フローチャートである。

【図6】ソフト同期とハード同期のタイムチャートである。

【図 7】従来の高信頼性情報処理装置の例を示す図である。

【符号の説明】

1 a ~ 1 n 处理装置

3 バス

4 ハード同期用レジスタ

5 生存処理装置表示フラグ

6 リプライ抑止回路

7 ソフト同期用フラグ

8 ソフト同期用レジスタ

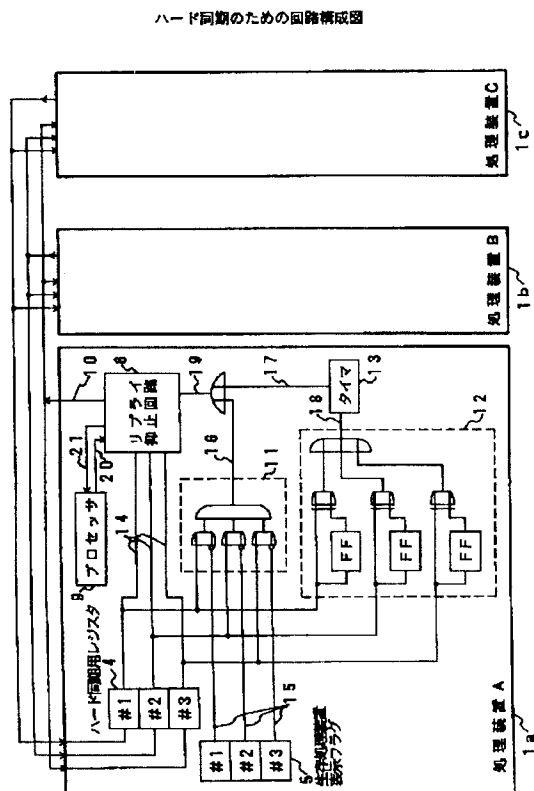
9 プロセッサ

20 10 リプライ抑止回路ビジー信号

1.4 ハード同期用レジスタ出力信号

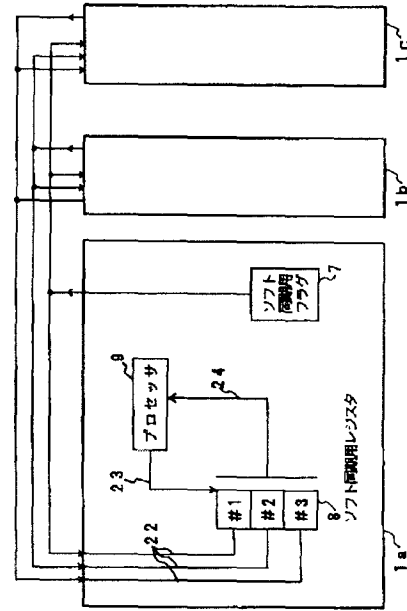
1 5 生存处理装置表示信号

【図2】

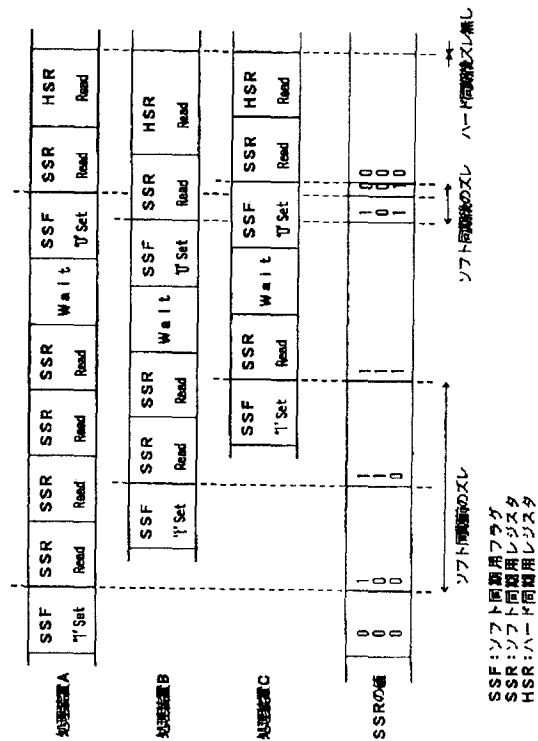


【図 4】

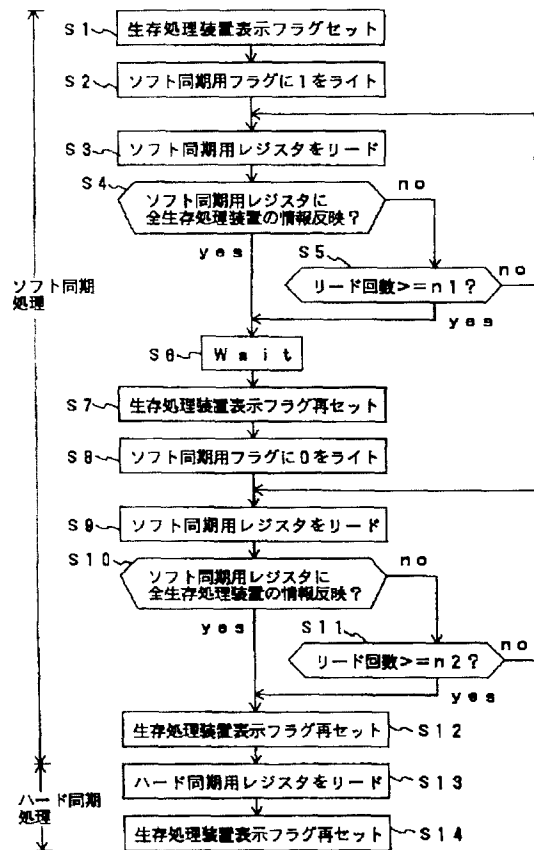
ソフト同期のための回路構成図



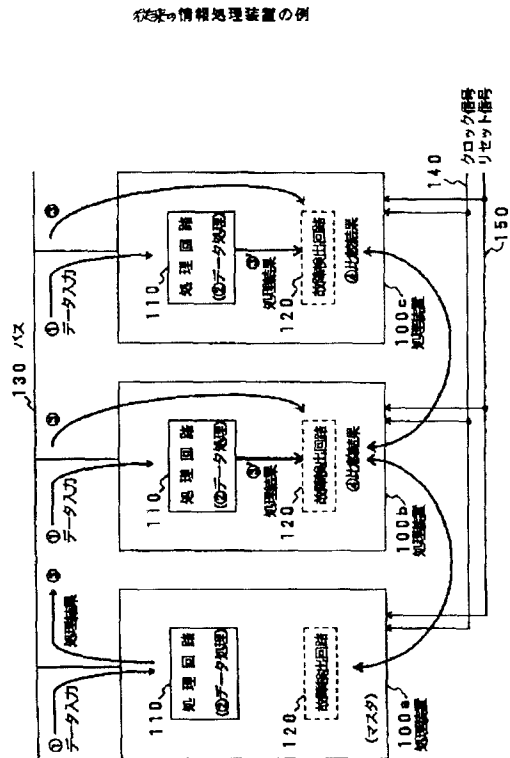
タイムチャート



【図5】



【図7】



フロントページの続き

(72) 発明者 加藤 慎哉
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 竹下 克典
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 菅野 文武
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 後藤 誠司
福岡県福岡市博多区博多駅前3丁目22番8
号 富士通九州デジタル・テクノロジー株
式会社内